

### **REMARKS/ARGUMENTS**

The title has been amended as requested by the Office Action. As to the objection and rejections with regard to claims 3 and 18 it is respectfully noted that the language “at least one compiler-generated operator” is different than the recited “compiler-generated code” of claim 1 and thus the requested correction has not been made. For the same reason, it is submitted that the §112 rejection is improper.

As to the rejection of various claims under 35 U.S.C. §102(e) over U.S. Publication No. 2004/0115860 (Johnson) the rejection is respectfully traversed. In this regard, the Johnson publication is an erroneous publication made by the Patent Office, in that the Specification printed is the Specification of the current application, as clearly seen in the image file wrapper (IFW) of the PTO for the Johnson publication. *See* original Specification of application 10/319,741 (the application number of the Johnson publication), which is entirely different than that published. Instead, the published Specification corresponds to the present application. Thus this rejection cannot stand.

Pending claims 1-5 and 16-19 stand rejected under 35 U.S.C. § 103 over U.S. Patent No. 6,931,634 (Croix) in view of U.S. Patent No. 5,339,419 (Chan). Applicants respectfully traverse the rejection. As to claim 1, the cited art nowhere teaches or suggests dynamically obtaining one or more program operators in a pre-compiler of a compiler module and applying a data transformation in the pre-compiler. Accordingly, claim 1 and the claims depending therefrom are patentable.

With regard to independent claim 16, the primary reference Croix nowhere teaches that a data transformation is applied to a portion of source code. Instead in Croix, an entire source code is data transformed. Further, as conceded by the Office Action, Croix nowhere teaches or suggests either dynamically obtaining of one or more program operators, or applying data transformation based on such operators. Nor does the secondary reference, Chan. In this regard, the Office Action refers to column 11, lines 37-46 and column 12, line 67 – column 13, line 20. These two passages from Chan are reproduced here:

The HPcode-Plus compiler intermediate language is an improvement upon HPcode, which was an improvement upon U-Code. U-Code is a compiler intermediate language which was originally used for distributing a Pascal compiler to the CRAY-1 and S-1 computer platforms. It was developed by

Stanford and the University of California at San Diego. U-Code, however, was not architecture neutral and could only support Pascal and Fortran source language programs.

Chan, col. 11, lns. 37-46.

The HPcode-Plus compiler intermediate language is very similar to assembly language for a HPcode-Plus virtual (i.e., fictional ) computer platform. Correspondingly, compiler intermediate representations written in the HPcode-Plus compiler intermediate language are very similar to assembly language programs for the HPcode-Plus virtual computer platform. Referring to FIG. 2, for example, the ANDF Producer 208 translates the source code 202 into its equivalent assembly language representation 212. The ANDF Producer 208, however, does not generate the assembly language representation 212 for its native computer platform 206. Rather, the ANDF Producer 208 generates the assembly language representation 212 for the HPcode-Plus virtual computer platform. The HPcode-Plus compiler intermediate language represents the assembly language for the HPcode-Plus virtual computer platform. The ANDF Installer 218, 228 receives the compiler intermediate representation 212, which represents assembly language for the HPcode-Plus virtual computer platform, and generates the object code 222, 232 for its target computer platform 216, 226.

Chan, col. 12, ln. 67 – col. 13, ln. 20.

As clearly seen, neither of these portions anywhere teach or suggest applying data transformation based on program operators. Instead all these portion and the remainder of Chan teach is that a source code is translated into a compiler intermediate language. Nothing, however, anywhere teaches or suggests applying data transformation based on program operators. For at least this reason claim 16 and the claims depending therefrom are patentable.


The remaining dependent claims that stand rejected under §103 over these references in further view of Johnson are overcome for the same reasons discussed above.

In view of these remarks, the application is now in condition for allowance and the Examiner's prompt action in accordance therewith is respectfully requested. The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 20-1504.

Respectfully submitted,

Date: \_\_\_\_\_

5/18/07



Mark J. Rozman  
Registration No. 42,117  
TROP, PRUNER & HU, P.C.  
1616 S. Voss Road, Suite 750  
Houston, Texas 77057-2631  
(512) 418-9944 [Phone]  
(713) 468-8883 [Fax]  
Customer Number: 21906